

本周周报（5.18-5.24）

刘昊南

本周工作

1. 本周主要工作为测试确定数据库选型，详细学习了 MongoDB 的搭建和使用方式，设计了出租车数据在 MongoDB 内存储的文档结构，同时利用了斐然师兄提供的温州一个月的出租车数据进行了测试
2. 我在自己的机器上搭建起了 MongoDB 服务器，数据在 MongoDB 内是以 Document 的方式存储的，每个 Document 相当于关系型数据库中的一条 Record，Document 聚合形成 Collection，每个 Collection 相当于关系型数据库中的 Table，但是一个 Collection 中的 Document 的字段可以不相同，即没有固定的 Schema。我目前采用的出租车数据的 Document 结构如下：

```
{
  "_id" : ObjectId<"555c83bfc6831513c046dde8">,
  "plateNumber" : "浙C02668",
  "location" : [
    120.59850311279297,
    28.026180267333984
  ],
  "time" : ISODate<"2013-12-31T16:00:31.976Z">,
  "isPassengerIn" : true,
  "speed" : 12,
  "direction" : 6
}
```

3. 在出租车数据的 Collection 中，我在 plateNumber、location 和 time 三个字段上建立了索引，以支持联合这三个属性的快速查询。其中，location 采用的是 MongoDB 所支持的 GeoSpatial 2dsphere 索引，支持地球表面经纬坐标系下的包含、相交、近邻关系的快速查询，结合 time 上的索引可以在时空上快速查询。
4. 针对斐然师兄提供的数据，编写了 Java 程序解析出租车数据并利用 MongoDB 的 Java driver 将数据存入了 MongoDB
5. 数据的时间范围为 1 个月，地理范围为温州市，二进制数据大小为 13G，测试结果如下：
 - a) 插入数据的时间没有精确测量，大致花费了不到 4 个小时的时间，一共插入了超过 3 亿条记录，属于正常范围
 - b) 建立索引花费了 3 个小时，由于数据量大，平均每个索引需要 1 个小时
 - c) 测试了车牌号与时间联合查询、经纬区间与时间联合查询，均可以做到瞬间得出结果
 - d) 存储性能较低，13G 的二进制数据插入之后占用了 89G 的硬盘空间，同

时 3 个字段上的索引占用了 43G 的硬盘空间，加起来占用了 130G 的空间，是原来的 10 倍。

6. 原因分析：由于 MongoDB 的 Document 是无模式的，所以对于 Document 的每个字段都要存储字段的名称、类型和大小，这里元信息占用的空间比数据的空间还要大。同时每个 Document 的大小如果不是 2 的幂，MongoDB 会将其填充为 2 的幂，也浪费了许多空间。MongoDB 不仅占用了大量磁盘空间，在提供服务时还会占用大量内存，但是也带来的写入和查询的高效率，之前斐然师兄在 Oracle 中做查询需要 20 秒。

下周计划

1. 跟斐然师兄讨论之后，觉得 MongoDB 占用的空间过大，如果是手机基站的数据绝对会硬盘不够用，目前准备往两个方面尝试
 - a) 缩短 Document 中字段的名称的长度，同时调整 MongoDB 填充 padding 的方式，来改善存储效率，做进一步的测试看看能达到什么程度，如果能将存储空间减少一半以上，同时将 MongoDB 部署到集群上应该可以硬盘应该可以满足需求
 - b) 考虑 Hadoop 集群的分布式文件系统 HDFS，以及基于 HDFS 的 BigTable 数据库 HBase，HBase 的存储效率要比 MongoDB 高，目前也有一些基于 HBase 的开源 GIS 索引准备详细了解